Road crack detection using a single stage detector based deep neural network

Carr, Thomas Arthur; Jenkins, Mark David; Iglesias, Maria Insa; Buggy, Tom; Morison, Gordon

Published in:
2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)

DOI:
10.1109/EESMS.2018.8405819

Publication date:
2018

Document Version
Peer reviewed version

Road Crack Detection using a Single Stage Detector Based Deep Neural Network

Thomas Arthur Carr*, Mark David Jenkins*, Maria Insa Iglesias*, Tom Buggy* & Gordon Morison*
*School of Engineering and Built Environment
Glasgow Caledonian University
Scotland, United Kingdom
Email: gordon.morison@gcu.ac.uk

***Abstract*—Condition and deterioration of public and private infrastructure is an issue that directly affects the majority of the world population. In this paper we propose the application of a Residual Neural Network to automatically detect road and pavement surface cracks. The high amount of variance in the texture of the surface and variation in illumination levels makes the task of automatically detecting defects within public and private infrastructure a difficult task. The system developed utilises a feature pyramid core with an underlying feed-forward ResNet architecture. The output from the feature pyramid then feeds into two sub-networks. One sub-network associates a class with the output from the feature pyramid. The other sub-network regresses the offset from each of the output bounding boxes of the feature pyramid to the corresponding ground truth boxes during training. The network was trained on real world data from an already established dataset. The data used to train and test on is very limited, due to the lack of available road crack datasets in the public domain. Despite the limited amount of data, the proposed method achieves a very positive results with minimal error.**

I. Introduction

The task of monitoring public infrastructure has traditionally been carried out by trained engineers and technicians. Over time, as infrastructure ages, the condition of that infrastructure steadily declines and the volume and severity of defects increases. This issue results in an increasing workload for the engineers and technicians, becoming both very expensive and time consuming [1]. Therefore as time passes the need for automation within this area only increases [2] [3].

In recent years, the shift of processing power from high end CPU's to GPU's, alongside the increasing availability of GPU technology, has resulted in new opportunities for the application of Machine Learning in the automation of manual tasks. Machine Learning, Deep Learning in particular, is rapidly increasing in popularity for industrial applications [4] [5]. Typically, Deep Learning Neural Networks require large amounts of training data to be effective. On a CPU this training would take many times longer than on a GPU which can run many operations in parallel. This shift in focus to GPU computing has resulted in Deep Learning applications that can be applied at a much lower cost and with a more realistic time of execution [6] [7] [8] [9].

One such application for Machine and Deep Learning is the assessment of assets for both public and private infrastructure, specifically the analysis of cracks in both road and pavement surfaces. The current method of assessing these assets is manual visual inspection, utilising techniques which are well established and documented. A number of existing applications which measure cracks within road surfaces are image processing and machine vision algorithms which segment the image to highlight the cracks. The existing segmentation algorithms [10] [11] [12] [13] [14] work by analysing an image pixel by pixel and assigning a probability distribution to each pixel. This distribution allows each pixel to be assigned for a particular class. The utilisation of Machine Learning techniques and algorithms to road crack segmentation is a new application with a limited number of papers published to date [15] [16] [17] [18]. The segmentation of cracks is both time and computationally intensive. A more computationally effective method is to detect the areas likely to contain cracks, therefore reducing the volume of data for the segmentation network. Detecting an area that is likely to contain a crack within both road and pavement surfaces using a detection based Neural Network is a new application with a limited number examples to compare to. The detection based approach is at its very core completely different to the segmentation based approach which identifies individual pixels that are likely to contain a crack.

Detection based Neural Networks have seen an influx of development in the past few years due to the ImageNet Detection competition [19]. Within the ImageNet Detection competition, the most popular underlying architecture is ResNet [20], which produces higher accuracy than many other architectures [21] [22] [23] [24] while being less computationally intensive. ResNet typically requires large amounts of data for training to develop an effective model. The larger amounts of data needed for ResNet mean that it is not ideal for this application due to the limited data, however, a variant of ResNet called RetinaNet [25] can be used on smaller datasets due to its optimised architecture. RetinaNet is based on a Feature Pyramid Network [26] (FPN) core which essentially scales the input to allow detection to take place at multiple scales. The feature pyramid is constructed around the feed-forward ResNet architecture.

This paper presents an algorithm for object detection of cracks within both road and pavement surfaces. The CrackForest dataset [10] [11] which consists of 118 images is utilised to train, validate and test the network. The CrackForest dataset was created for a segmentation based approach and therefore

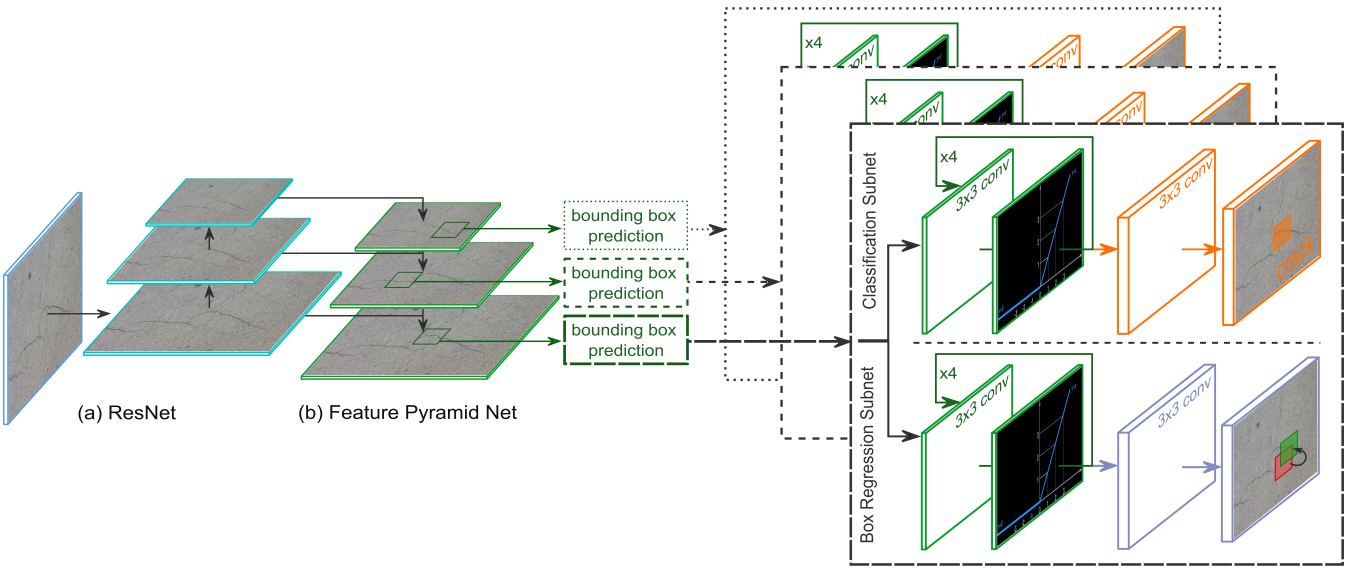

Fig. 1. **Structural Overview of RetinaNet.** RetinaNet is a one-stage detection based Neural Network based on an FPN [26] which is built on top of a feed-forward ResNet architecture shown with label a). The FPN shown with label b) returns an output of anchor boxes which are the coordinates of the areas of interest, these are then passed into two separate sub-networks. One sub-network classifies the anchor boxes and the other offsets them to correct any positional errors from the FPN. The output of both sub-networks is fed directly into the focal loss function which fits the network to harder or more sparse examples.

does not contain bounding box style labels. The CrackForest dataset was acquired by the researchers [10] [11] from the real world with an ordinary iPhone camera and the segmentation masks were labelled pixel by pixel. The detection based approach operates on images which have been labelled with bounding box class identifiers. As these labels do not exist within the CrackForest dataset they have been created manually.

This detection network is intended to be the first stage in a surface crack analysis pipeline, the second stage of which will be semantic segmentation. Performing bounding box detection first will allow the segmentation network to be focused on areas of interest, reducing the overall computation required.

II. Network Architecture

As described above, the Neural Network RetinaNet is well suited for the detection of cracks in road and pavement surfaces. RetinaNet was designed to improve the accuracy of one stage detectors. The accuracy increase without a major time penalty is achieved by splitting the network following the FPN into two separate sub-networks. These two sub-networks are executed in parallel, with the same input layers, making the network faster to execute. The utilisation of two sub-networks simulates a typically more accurate two-stage detector like Faster R-CNN with FPN [26]. Alongside the reduced time and increased accuracy compared to other methods, RetinaNet [25] addresses the class imbalance in training by reshaping the loss function to down-weight easy or overweighted class examples. This down-weighting of overweighted class examples correlates to the network learning more from the harder examples than it would in any other Neural Network. This optimisation skews the network to consider all of the examples and try to learn more from the sparser examples to achieve more accurate real world results.

The network utilised within RetinaNet is based upon a ResNet architecture with FPN [26] built on top. The FPN architecture constructs multiple scaled images from the single input image. This scaling allows the detection of objects in a more robust, scale invariant way. Each of the levels of the FPN produced is effectively a standard convolution, each of which are evaluated independently by the detector which predicts bounding boxes at each scale. The process for identifying these prediction (anchor) regions consists of a sliding window that includes multiple scales of predictions for each level of the FPN. For each of the multiple scaled predictions, three different aspect ratios are used (2:1, 1:1, 1:2). The anchor area applied varies in size based on the level of the FPN, from 32^2 at FPN level 3 to 512^2 at FPN level 7, as displayed in Figure 1. During training the weighting of the FPN is determined by the Intersection over the Union (IoU) ratio between the predicted output and the ground-truth bounding box. If the IoU is greater than 0.5 then the match is considered positive, provided that the anchor predicted is assigned to only one ground-truth box. Any overlap between multiple predictions and ground-truth boxes is ignored during training. The scale of the ground-truth boxes does not vary with the FPN scale, instead the anchor boxes are utilised to scale the prediction back to the original image scale for comparison to the ground-truth boxes.

The classification subnet calculates the probability that the output of the FPN contains a crack and assigns the appropriate

class label. The subnet operates on each of the FPN levels with the parameters shared across all subnet levels. At each of the levels, the subnet applies a small Fully Convolutional Network (FCN) as displayed in Figure 1.

During training the box registration subnet correlates each anchor box to a corresponding nearby ground-truth bounding box, provided that one exists. The design of this network is the same as the classification subnet except that it is executed four times for each of the predicted boxes, to calculate the offset scale and positioning. The four outputs predict the offset between the predicted box and the ground-truth. For each of the anchors, these four linear outputs predict the relative offset from the ground-truth objects, two for the scaling of the bounding box and two for the positioning of the bounding box.

The two sub-networks, then feed into the focal loss function. The focal loss function focuses on the outliers and misclassified examples by weighting these examples higher in the focal loss function. This weighting means that these harder examples have a greater effect on the network as a whole and mean that variations within classes are accounted for. The weighting is achieved by adjusting a modulating factor, which essentially adjusts the amount of focus on outlier results, resulting in the harder examples being more visible in the loss function.

III. Evaluation and Results

To evaluate the network, RetinaNet was trained on road crack images from the CrackForest dataset [10] [11] which consists of 118 single channel grey-scale images (80 train, 20 validation and 18 test) of size [480x340]. The dataset is supplied with corresponding binary ground truth masks but no bounding box coordinates, as CrackForest was designed for segmentation rather than detection. For a detection based network like RetinaNet, bounding box labels are needed, including co-ordinates within the image and a class identifier. In the case of CrackForest, there was a single class in the dataset (cracks), so all of the cracks within the images were labelled with bounding boxes of this class.

To allow for a fair evaluation of the algorithm against the segmentation masks provided within CrackForest [10] [11], a similar evaluation technique was utilised. The detection based Neural Network produces an output of bounding box coordinates. These bounding boxes were compared directly to the CrackForest ground-truth segmentation masks.

This means that the true positives are the total number of bounding boxes that contain a crack. The false positives are the total number of bounding boxes which do not contain a crack. False negatives are defined as crack pixels in the segmentation masks which are not detected by a bounding box. The first performance metric selected is precision (Equation 1) where Pr is the precision measured as a percentage, TP is the true positives or the correctly detected boxes and FP is the false positives which are the predicted boxes which do not contain cracks. The second performance metric is the percentage of a segmentation mask not correctly detected (Equation 2), where Es is the error in segmentation measured as a percentage, Ep is the total number of incorrectly detected pixels and Cp is the total number of correctly detected pixels. This metric has been selected due to the overarching goal of feeding the output into a segmentation based Neural Network. Therefore it is essential to consider the output in terms of pixels missed. Another performance metric to consider is the percentage reduction of data which will ultimately be passed to the segmentation network. This performance metric is useful for determining the advantage of utilising a detection based network before a segmentation based network, as such Equation 3 details how this reduction is calculated. Within Equation 3 PeR is the percentage reduction, Td is the total number of pixels detected and Tp is the total number of pixels in the image.

$$Pr = \frac{TP}{TP + FP} \tag{1}$$

$$Es = \frac{Ep}{Ep + Cp} \tag{2}$$

$$PeR = \frac{Td}{Tp} \tag{3}$$

Training was completed on a Titan Xp GPU with 12GB of RAM. The implementation of the Neural Network was executed by Keras and Tensorflow with the number of training epochs at 50 with a batch size of 1. A snapshot of the model was saved after each epoch, to allow back tracking in case of over-fitting. Training takes 90 minutes per epoch and the average time taken for inference on the CrackForest dataset images is 0.0531 seconds. The proposed network achieves a Precision of 98.92% and a segmentation error of 1.22%. These results show a very strong overlap between the prediction bounding boxes and the ground-truth bounding boxes, which is reflected in the high precision and low segmentation error percentage. As per the RetinaNet [25] paper, only those bounding boxes with a confidence of greater than 50% were considered and displayed. Considering this network as a precursor for a segmentation based network, the high precision and recall mean that the segmentation based network will spend less computational cost evaluating areas that are not of interest. A segmentation network would need to run on all of the pixels within the image, therefore the detection based network offers significant speed up by providing coordinates for the segmentation based network to run on. Application of this detection network results in a total reduction of 72.12% in the volume of data passed to the segmentation network. Therefore the combination of the detection and segmentation based networks will result in faster execution times that may tend towards a more real time application.

A selection of the images gained from these results are shown in Figure 2 with both bounding boxes and an overlaid heat-map of the results. In Figure 2, label a) represents the original image, label b) represents the bounding box highlighted images, label c) is simply a heat-map applied to the confidence ratings of each bounding box, label d) shows the segmentation masks and label e) shows the true positives in green, the false

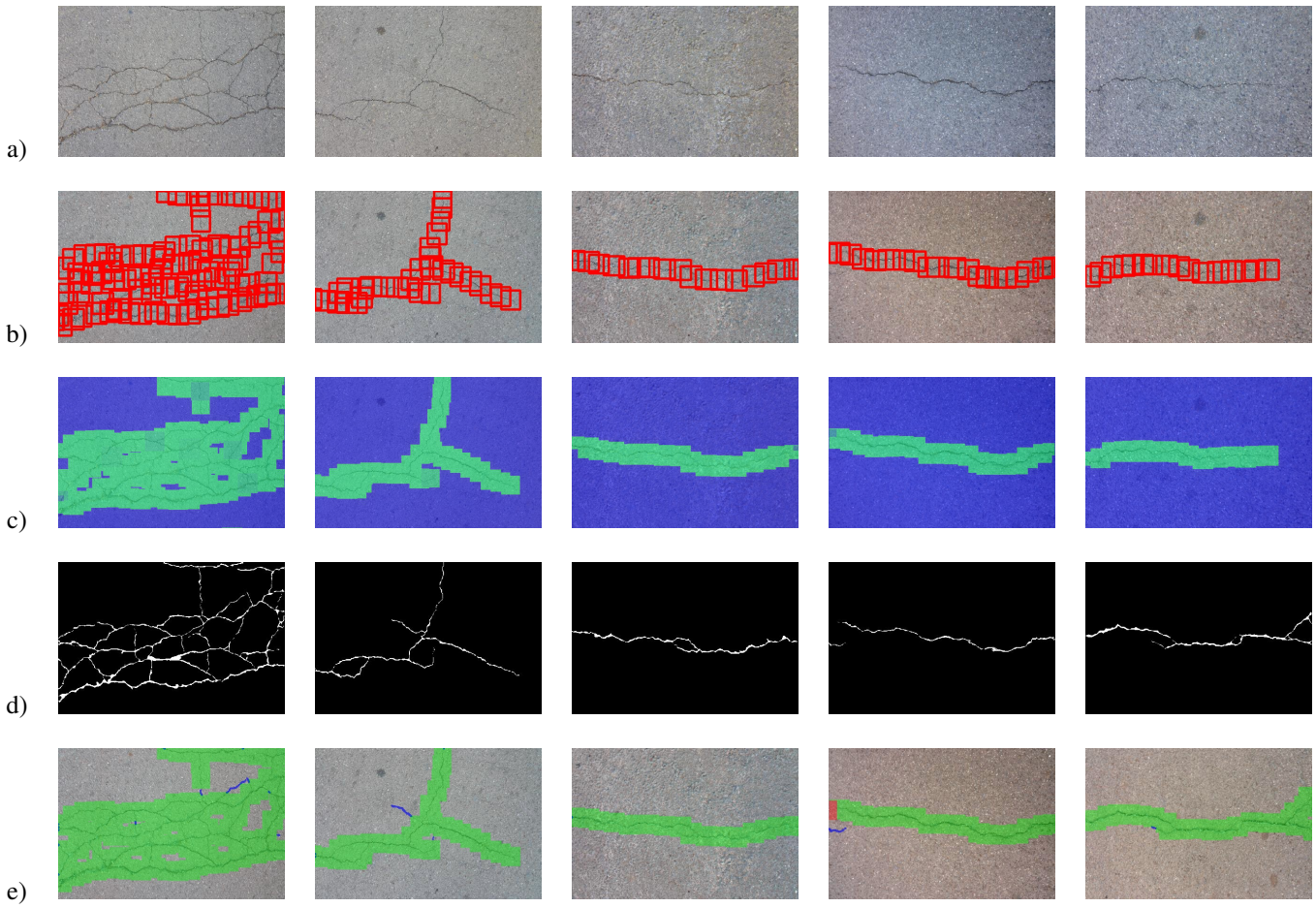

Fig. 2. **Results of Inference on Subsection of Test Images.** Where label a) represents the input image. Label b) represents the image with the predicted bounding boxes superimposed on top of the image. Label c) represents the certainty of each prediction colour mapped and then superimposed on top of the input image. Within the colour map, green represents greater than 99% certain, blue represents no prediction and the darker green colours represent between 99% and 50% certain. Label d) shows the segmentation mask used to evaluate the predictions for each image. Label e) shows a map of the performance metrics super imposed onto the input image. Within this mapping, the correctly detected bounding boxes are highlighted in green, the incorrectly detected bounding boxes are highlighted in red and the crack pixels missed are highlighted in blue

positives in red and the false negatives highlighted in blue. The row labelled e) within Figure 2 shows how successful the network is with only a few pixels not detected correctly and little to no false positives.

IV. Conclusion

Within the field of structural monitoring, the need for automation is increasing as both public and private infrastructure age and deteriorate. The detection of cracks in road and pavement surfaces is just one of a number of areas to start benefiting from the GPU implementation of Machine Learning algorithms. The work presented in this paper focuses on the detection of cracks within images of both road and pavement surfaces. The Neural Network utilised is called RetinaNet. RetinaNet is based on a Feature Pyramid Network built on top of ResNet in order to accomplish the accuracy of a two stage detector within a single stage detector. The results show a very high precision score and a low segmentation percentage error given the very limited dataset trained upon. The high percentage reduction in area detected by RetinaNet means that it is ideally suited to be the first stage in a surface crack analysis pipeline, followed by a segmentation based Neural Network for pixel by pixel analysis.

The speed of the network is a key component to consider for the goal of feeding the output into a segmentation based network. The fact that RetinaNet [25] is a one-stage detector with the accuracy of a two stage detector made it ideal for this task, with an average execution time of 0.0531 seconds for inference when running on the CrackForest dataset [10] [11].

Given that the network selected concentrates on hard examples, any class imbalances are addressed automatically. In the future as more data becomes available the detection based approach will expand to further classes to include features such as Potholes and Raveling. Alongside adding

extra classes to the network, the errors that the network produced will be refined and the results will be improved.

ACKNOWLEDGEMENTS

The work presented in this paper has been carried out in association with Geckotech Solutions Ltd who provide specialist access solutions to the railway and construction engineering industries. The authors would like to thank Geckotech Solutions Ltd for their support.

REFERENCES

[1] S. C.Radopoulou I. Brilakisb "Improving Road Asset Condition Monitoring." *Transportation Research Procedia Vol. 14, 2016, p. 3004-3012*

[2] M. D. Jenkins, T. Buggy and G. Morison "An Imaging System for Visual Inspection and Structural Condition Monitoring of Railway Tunnels." *2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS), Milan, 2017, pp. 1-6.*

[3] A. L. Pyayt et al. "Artificial intelligence and finite element modelling for monitoring flood defence structures." *2011 IEEE Workshop on Environmental Energy and Structural Monitoring Systems, Milan, 2011, pp. 1-7.*

[4] M. Gallo, F. Simonelli, G. De Luca and C. Della Porta, "An artificial neural network approach for spatially extending road traffic monitoring measures." *2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS), Bari, 2016, pp. 1-5.*

[5] M. De Nadai and M. van Someren, "Short-term anomaly detection in gas consumption through ARIMA and Artificial Neural Network forecast." *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings, Trento, 2015, pp. 250-255.*

[6] D. Steinkraus, I. Buck and P.Y. Simard "Using GPUs for machine learning algorithms." *ICDAR '05 Proceedings of the Eighth International Conference on Document Analysis and Recognition. p. 1115-1119*

[7] K. Chellapilla, S. Puri, P. Simard "High Performance Convolutional Neural Networks for Document Processing." *In 10th International Workshop on Frontiers in Handwriting Recognition, 2006.*

[8] R. Raina, A. Madhavan and A. Y. Ng "Large-scale deep unsupervised learning using graphics processors." *Proc. 26th Annual International Conference on Machine Learning p. 873 - 880 (2009).*

[9] D. Ciresan, U. Meier and J. Schmidhuber "Multi-column Deep Neural Networks for Image Classification." *CVPR '12 Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 3642-3649*

[10] S. Yong, C. Limeng, Q. Zhiquan, M. Fan and C. Zhensong Automatic Road Crack Detection Using Random Structured Forests. *IEEE Transactions on Intelligent Transportation Systems, Vol.17, No.12, pp 343 - 445, 2016.*

[11] C. Limeng, Q. Zhiquan, C. Zhensong, M. Fan and S. Yong Pavement Distress Detection Using Random Decision Forests. *ICDS 2015 Proceedings of the Second International Conference on Data Science - Volume 9208 Pages 95-102.*

[12] H. Oliveira and P. L. Correia CrackIT An Image Processing Toolbox For Crack Detection and Characterization. *2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 798-802. doi: 10.1109/ICIP.2014.7025160.*

[13] H. Oliveira and P. L. Correia Automatic Road Crack Detection and Characterization. *EEE Transactions on Intelligent Transportation Systems, Vol.14, No.1, pp. 155168, Mar. 2013.*

[14] A. Rabih, C. Sylvie, I. Jrme and V. Baltazart Automatic Crack Detection on Two Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection." *IEEE Transactions on Intelligent Transportation Systems, Vol.17, No.10, pp 2718 - 2729, 2016*

[15] Z. Fan, Y. Wu, J. Lu and W. Li "Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network." *arXiv preprint 2018, arXiv:1802.02208*

[16] L. Pauly, D. Hogg and R. Fuentes "Deeper Networks for Pavement Crack Detection." *In: Proceedings of the 34th ISARC. 34th International Symposium in Automation and Robotics in Construction, 28 Jun - 01 Jul 2017, Taipei, Taiwan. IAARC , pp. 479-485*

[17] L. Zhang, F. Yang, Y. Daniel Zhang and Y. J. Zhu "Road Crack Detection using Deep Convolutional Neural Network." *2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016, pp. 3708-3712*

[18] S. Yokoyamaa and T. Matsumotob "Development of an Automatic Detector of Cracks in Concrete Using Machine Learning" *Procedia Engineering Volume 171, 2017, Pages 1250-1255*

[19] O. Russakovsky, J. Deng, H. Su, et al ImageNet Large Scale Visual Recognition Challenge." *Int J Comput Vis (2015) 115: 211. https://doi.org/10.1007/s11263-015-0816-y*

[20] K. He, X. Zhang, S. Ren and J. Sun "Deep Residual Learning for Image Recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778. doi: 10.1109/CVPR.2016.90*

[21] A. Krizhevsky, I. Sutskever and G. E. Hinton "ImageNet classification with deep convolutional neural networks." *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. p. 1097-1105*

[22] K. Simonyan and A. Zisserman "Very Deep Convolutional Networks for Large-Scale Image Recognition." *2014 eprint arXiv:1409.1556*

[23] C. Szegedy et al "Going deeper with convolutions." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.*

[24] S. Ren, K. He, R. Girshick, J. Sun "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. Pages 91-99.*

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar "Focal Loss for Dense Object Detection." *arXiv preprint arXiv:1708.02002, 2017.*

[26] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan and S. Belongie "Feature Pyramid Networks for Object Detection." *eprint arXiv:1612.03144*

[27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman "The PASCAL Visual Object Classes (VOC) Challenge." *Int J Comput Vis (2010) 88: 303. https://doi.org/10.1007/s11263-009-0275-4*

[28] Lin TY. et al. "Microsoft COCO: Common Objects in Context." *Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham*